2. Growth of Function

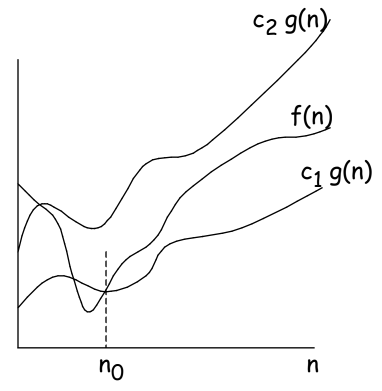
- Typically, problems become computationally intensive as the input size grows.
- We look at input sizes large enough to make only the order of the growth of the running time relevant for the analysis and comparison of algorithms.
- Hence we are studying the <u>asymptotic</u> efficiency of algorithms.
- So far our analysis showed that:
  Merge-Sort has a running time of Θ (n lg n)
  Insertion-Sort has a running time of Θ (n^2)
- We like to make this notion more precise.

Θ-Notation

- Definition: Let g(n) be an asymptotically non-negative function on the natural numbers.

$$\Theta(g(n)) = \{f(n) \mid \quad c_1 > 0, c_2 > 0, n_0 \quad \text{Nat} \bullet \quad n \geq n_0 \bullet 0 \leq c_1\, g(n) \leq f(n) \leq c_2\, g(n)\}$$

- Function f(n) belongs to $\Theta(g(n))$ if it can be sandwiched between c_1 g(n) and c_2 g(n) for some constants c_1, c_2, for all n greater than some n_0.
- In this case, we say that g(n) is an <u>asymptotically tight bound</u> for f(n).



- We write f(n) = $\Theta(g(n))$ for f(n) $\Theta(g(n))$

Examples for Θ …

- $n^2 / 2 - 3 n = \Theta(n^2)$
  We have to determine $c_1 > 0$, $c_2 > 0$, n_0 Nat such that:
  $c_1 n^2 \le n^2 / 2 - 3 n \le c_2 n^2$ for any $n \ge n_0$
  Dividing by n^2 yields:
  $c_1 \le 1 / 2 - 3 / n \le c_2$
  This is satisfied for $c_1 = 1 / 14$, $c_2 = 1 / 2$, $n_0 = 7$.

- $6 n^3 \ne \Theta(n^2)$
  We would have to determine $c_1 > 0$, $c_2 > 0$, n_0 Nat such that:
  $c_1 n^2 \le 6 n^3 \le c_2 n^2$ for any $n \ge n_0$
  which cannot exist.

31

… Examples for Θ

- $a n^2 + b n + c = \Theta(n^2)$ provided $a > 0$
  For example, take $c_1 = a / 4$, $c_2 = 7 a / 4$,
  $n_0 = 2 \max (|b|/a, \sqrt{(|c|/a)})$.

- In general, if $a_m > 0$ then

$$\sum_{i=0}^{m} a_i n^i = \Theta(n^m)$$

32

Properties of Θ

- Assume f(n) and g(n) are asymptotically positive:
- $f(n) = \Theta(g(n))$ $g(n) = \Theta(h(n))$ $f(n) = \Theta(h(n))$ (Transitivity)
- $f(n) = \Theta(f(n))$ (Reflexivity)
- $f(n) = \Theta(g(n)) \Leftrightarrow g(n) = \Theta(f(n))$ (Symmetry)
- $\max(f(n), g(n)) = \Theta(f(n) + g(n))$ (Maximum)
  If f(n) and g(n) are the running times of the two branches of an if-statement, then this can be used to get a tight bound on the worst case running time of the whole statement if nothing is known about the condition, e.g. depends on unknown input.

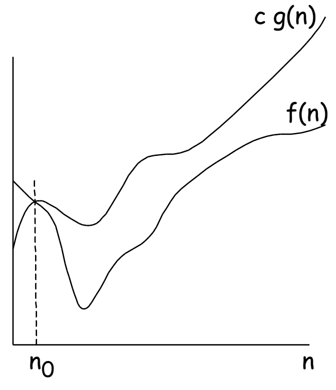
33

O-Notation

- **Definition**: Let g(n) be an asymptotically non-negative function on the natural numbers.

  $O(g(n)) = \{f(n) \mid \quad c > 0, n_0 \quad \text{Nat} \cdot$
  $n \geq n_0 \cdot 0 \leq f(n) \leq c\, g(n)\}$

- In this case, we say that g(n) is an asymptotic upper bound for f(n).
- Θ is stronger than O:
  $f(n) = \Theta(g(n)) \quad f(n) = O(g(n))$,
  or $\Theta(g(n)) \quad O(g(n))$
- We write $f(n) = O(g(n))$ for $f(n) \quad O(g(n))$



34

Examples for *O*

- $a\,n^2 + b\,n + c = O(n^2)$ provided $a > 0$ since it is also $\Theta(n^2)$.
- $a\,n + b = O(n^2)$ provided $a > 0$
- $n \lg n + n = O(n^2)$
- $\lg^k n = O(n)$ for all k Nat
- *O* can be used for an upper bound of the running time for worst-case input (and hence for any input).
- Note: Some books use *O* to informally describe tight bounds. Here we use Θ for tight bounds and *O* for upper bounds.

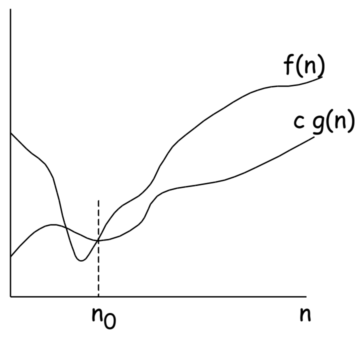
35

Ω-Notation

- **Definition**: Let g(n) be an asymptotically non-negative function on the natural numbers.

  $\Omega(g(n)) = \{f(n) \mid \quad c > 0,\ n_0 \quad \text{Nat} \cdot$
  $n \geq n_0 \cdot 0 \leq c\,g(n) \leq f(n)\}$

- In this case, we say that g(n) is an <u>asymptotic lower bound</u> for f(n).
- Θ is stronger than Ω:
  $f(n) = \Theta(g(n)) \quad f(n) = \Omega(g(n))$,
  or $\Theta(g(n)) \quad \Omega(g(n))$
- We write $f(n) = \Omega(g(n))$ for $f(n) \quad \Omega(g(n))$



36

Examples for Ω

- $a n^2 + b n + c = \Omega(n^2)$ provided $a > 0$ since it is also $\Theta(n^2)$.
- $a n^2 + b n + c = \Omega(n)$ provided $a > 0$
- Ω can be used for a lower bound of the running time for best-case input (and hence for any input). For example, the best-case running time of Insertion-Sort is $\Omega(n)$.

Properties of Θ, O, Ω

- f(n) is a tight bound if it is an upper and lower bound:
  $f(n) = \Theta(n) \Leftrightarrow f(n) = O(n) \quad f(n) = \Omega(n)$
- $f(n) = O(g(n)) \quad g(n) = O(h(n)) \quad f(n) = O(h(n))$
  $f(n) = \Omega(g(n)) \quad g(n) = \Omega(h(n)) \quad f(n) = \Omega(h(n))$ (Transitivity)
- $f(n) = O(f(n))$
  $f(n) = \Omega(f(n))$ (Reflexivity)
- $f(n) = O(g(n)) \Leftrightarrow g(n) = \Omega(f(n))$ (Transpose Symmetry)

Example for O, Ω, Θ

- $3n^2 - 100n + 6 = O(n^2)$ because $3n^2 > 3n^2 - 100n + 6$
- $3n^2 - 100n + 6 = O(n^3)$ because $.00001\,n^3 > 3n^2 - 100n + 6$
- $3n^2 - 100n + 6 \neq O(n)$ because $c\,n < 3n^2$ when $n > c$
- $3n^2 - 100n + 6 = \Omega(n^2)$ because $2.99\,n^2 < 3n^2 - 100n + 6$
- $3n^2 - 100n + 6 \neq \Omega(n^3)$ because $3n^2 - 100n + 6 < n^3$
- $3n^2 - 100n + 6 = \Omega(n)$ because $10^{10}\,n < 3n^2 - 100 + 6$
- $3n^2 - 100n + 6 = \Theta(n^2)$ because both O and Ω
- $3n^2 - 100n + 6 \neq \Theta(n^3)$ because not Ω
- $3n^2 - 100n + 6 \neq \Theta(n)$ because not O

39

Asymptotic Notation in Equations

- $f(n) = \Theta(n)$ simply means $f(n) \quad \Theta(n)$
- More generally, $\Theta(n)$ stands for an anonymous function which is an element of $\Theta(n)$, e.g.

  $3n^2 + 3n + 1 = 2n^2 + \Theta(n)$

  means

  $3n^2 + 3n + 1 = 2n^2 + f(n) \quad f(n) \quad \Theta(n)$ for some f
- In recurrences:

  $T(n) = 2\,T(n / 2) + \Theta(n)$
- In calculations:

  $$\begin{aligned} 2n^2 + 3n + 1 &= 2n^2 + \Theta(n) \\ &= \Theta(n^2) \end{aligned}$$

40

o-Notation

- The upper bound provided by O may or may not be tight. We use o for an upper bound which is not tight.

- Definition: Let g(n) be an asymptotically non-negative function on the natural numbers.

  $o(g(n)) = \{f(n) \mid \quad c > 0 \bullet \quad n_0 \quad \text{Nat} \bullet$
  $n \geq n_0 \bullet 0 \leq f(n) \leq c\, g(n)\}$

  The idea of the definition is that f(n) becomes insignificant relative to g(n) as n approaches infinity.

- For example:
  - $2\, n = o(n^2)$
  - $2\, n^2 \neq o(n^2)$
  - $2\, n^3 \neq o(n^2)$

ω-Notation

- The lower bound provided by Ω may or may not be tight. We use ω for a lower bound which is not tight.

- Definition: Let g(n) be an asymptotically non-negative function on the natural numbers.

  $\omega(g(n)) = \{f(n) \mid \quad c > 0 \bullet \quad n_0 \quad \text{Nat} \bullet$
  $n \geq n_0 \bullet 0 \leq c\, g(n) \leq f(n)\}$

  The idea of the definition is that g(n) becomes insignificant relative to f(n) as n approaches infinity.

- For example:
  - $n^2 / 2 = \omega(n)$
  - $n^2 / 2 \neq \omega(n^2)$
  - $n^2 / 2 \neq \omega(n^3)$

Example: Factorial

- n! is defined by:

  0! = 1

  n! = (n-1)! for n>0

- From Stirling's approximation

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1 + \Theta\left(\frac{1}{n}\right)\right)$$

  one can derive:
  - n! = o(n^n)
  - n! = ω (2^n)